IN THE SPECIFICATION

In accordance with 37 CFR § 1.121(b)(1), applicant provides the following replacement paragraph(s) and associated marked-up versions.

On page 1, please replace the paragraph beginning at line 16 and ending at line 21, with the following paragraph:

Page ~~1~~ 2, lines 16-21: Replacement Paragraph:

A2

The Advanced Configuration and Power Interface (ACPI) Specification 1.0b provides a uniform set of definitions, power management states, and the like for implementing power conservation features in a computing system. The ACPI Specification defines the S1 power state as a low latency power state in which all system context is maintained (see § 9.1.1).

Page 1, lines 16-21: Markup Paragraph:

The Advanced Configuration and Power Interface (ACPI) Specification 1.0b ~~(Revision 1.0b of this open industry specification is available at http://www.teleport.com/~acpi/)~~ provides a uniform set of definitions, power management states, and the like for implementing power conservation features in a computing system. The ACPI Specification defines the S1 power state as a low latency power state in which all system context is maintained (see § 9.1.1).

On page 3, please replace the paragraph beginning at line 5 and ending at line 12, with the following paragraph:

Page 3, lines 5-12: Replacement Paragraph:

One example of a memory architecture is the Rambus® technology available from Rambus Corporation of Mountain View, California. Some Rambus® memories offer various power conservation modes (see "Direct Rambus™ Memory for Mobile PCs" also available from Rambus Corporation). Active, nap, standby, and PwrDown (powerdown) modes are available. An RDRAM® device automatically transitions to standby mode at the end of a transaction. When a memory transaction request is sent out to the memory array, the appropriate RDRAM® device exits standby and services the request.

A3

Page 3, lines 5-12: Markup Paragraph:

One example of a memory architecture is the ~~Rambus™~~Rambus® technology ~~memory architecture~~ available from Rambus Corporation of Mountain View, California. Some Rambus™ ~~®~~ ~~parts~~ memories offer various power conservation modes (see "Direct Rambus™ Memory for Mobile PCs" also available from Rambus Corporation). Active, nap, standby, and PwrDown (powerdown) modes are available. ~~A Rambus™ Dynamic Random Access Memory (RDRAM)~~ An RDRAM® device automatically transitions to standby mode at the end of a transaction. When a memory transaction request is sent out to the memory array, the appropriate RDRAM® device exits standby and services the request.

On page 3, please replace the paragraph beginning at line 13 and ending at line 18, with the following paragraph:

Page 3, lines 13-18: Replacement Paragraph:

Power consumption may be further reduced by placing RDRAM® devices in a nap mode or a powerdown mode. Nap and powerdown modes may be entered by sending commands to the memory. From both the nap mode and the powerdown mode, a resynchronization time is required by the RDRAM® devices for memory system's delay locked loop to synchronize the interface to the channel clock.

A4

Page 3, lines 13-18: Markup Paragraph:

Power consumption may be further reduced by placing ~~RDRAMs~~ RDRAM® devices in a nap mode or a powerdown mode. Nap and powerdown modes may be entered by sending commands to the memory. From both the nap mode and the powerdown mode, a resynchronization time is required by the RDRAM® devices for memory system's delay locked loop to synchronize the ~~RDRAM~~ interface to the channel clock.

On page 10, please replace the paragraph beginning at line 10 and ending at line 17, with the following paragraph:

Page 10, lines 10-17: Replacement Paragraph:

For example, in a system which utilizes a Rambus® memory subsystem, putting the memory subsystem in a nap state or a powerdown state requires that certain initialization operations be performed prior to the memory subsystem resuming normal operations and returning any data from memory. Thus, if a system enters the ACPI S1 state and expects to exit by executing operating system code, the memory subsystem 160 will be unavailable to retrieve that operating system code. Thus, the intervention of hardware memory resume logic 130 may allow the ACPI S1 state to be used in a system with a Rambus® memory subsystem.

A5

Page 10, lines 10-17: Markup Paragraph:

For example, in a system which utilizes a ~~Rambus™~~ Rambus® memory subsystem, putting the memory subsystem in a nap state or a powerdown state requires that certain initialization operations be performed prior to the memory subsystem resuming normal operations and returning any data from memory. Thus, if a system enters the ACPI S1 state and expects to exit by executing operating system code, the memory subsystem 160 will be unavailable to retrieve that operating system code. Thus, the intervention of hardware memory resume logic 130 may allow the ACPI S1 state to be used in a system with a ~~Rambus™~~ Rambus® memory subsystem.

On page 10, please replace the paragraph beginning at line 18 and ending at page 11, line 6, with the following paragraph:

Page 10, line 18 – Page 11, line 6: Replacement Paragraph:

Details of the initialization operations performed in block 230 for one embodiment are shown in the flow diagram of Figure 3. In block 300, the memory interface control logic is initialized by the memory resume logic 130. In the embodiment of Figure 1, the memory interface control logic is the memory control logic 135. In embodiments utilizing a Rambus® memory subsystem, the memory control logic 135 may be a Rambus ASIC Cell (RAC) which may be initialized according to Rambus specifications (e.g., execute a RAC initialization operation and set control register values as needed). These operations are performed well after the appropriate clocks are running and stable. In block 310, the memory resume logic 130 sends a clock synchronization command and the system waits for memory subsystem clocks to synchronize. For example, the system may wait for the Direct Rambus Clock Generator (DRCG) to lock to the Clock To Memory (CTM) clock.

Page 10, line 18 – Page 11, line 6: Markup Paragraph:

Details of the initialization operations performed in block 230 for one embodiment are shown in the flow diagram of Figure 3. In block 300, the memory interface control logic is initialized by the memory resume logic 130. In the embodiment of Figure 1, the memory interface control logic is the memory control logic 135. In embodiments utilizing a ~~Rambus™~~Rambus® memory subsystem, the memory control logic 135 may be a Rambus ASIC Cell (RAC) which may be initialized according to Rambus specifications (e.g., execute a RAC initialization operation and set control register values as needed). These operations are performed well after the appropriate clocks are running and stable. In block 310, the memory resume logic 130 sends a clock synchronization command and the system waits for memory subsystem clocks to synchronize. For example, the system may wait for the Direct Rambus Clock Generator (DRCG) to lock to the Clock To Memory (CTM) clock.